

# Intelligent Access Control System For Safety Critical Areas

## *Introduction*

### **Overview:**

In some industries it is necessary for the workers to wear safety helmets and shoes while working. So to check whether workers are taking safety precautions or not we are proposing this system.

### **Purpose:**

We can train our classifier to identify helmet and safety shoes with IBM Cloud. There will be video streaming near the entry of the industries where we can first detect the face of a person and if any person is present then we can capture the image of that moment and send it to IBM Cloud to detect whether the person is wearing helmet or shoe.

If the person is wearing shoe and helmet we can give him access by opening the door. If he is not wearing then we can restrict his access by not opening the door. We can even warn him through voice commands to take the safety precautions

## *Literature Survey*

### **Existing problem:**

In some industries because of the violating the safety precautions, industries have been facing many accidents, which are leading to the loss of properties and lives of the people. So to overcome that issue, intelligent access control system for safety critical areas system could help. By allowing the person with proper safety equipment.

### **proposed solution:**

As to prevent these accidents and also for the safety purpose, this access control could help the user to monitor the people.

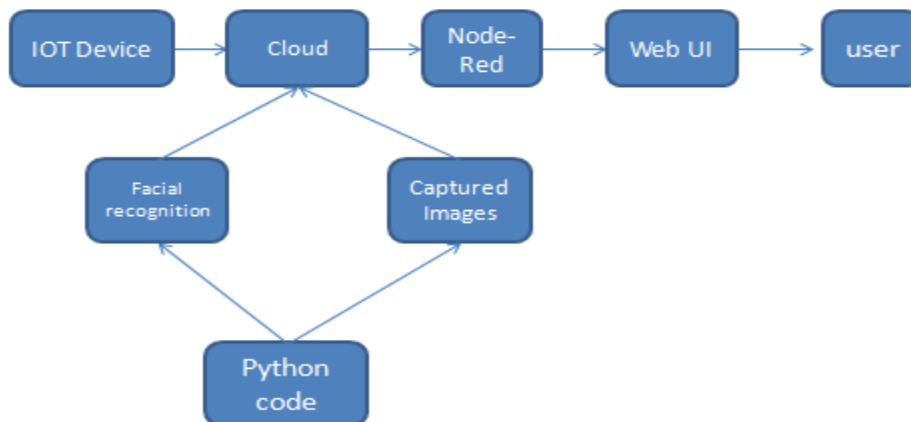
## *Theoretical Analysis*

### HARDWARE AND SOFTWARE DESIGN :

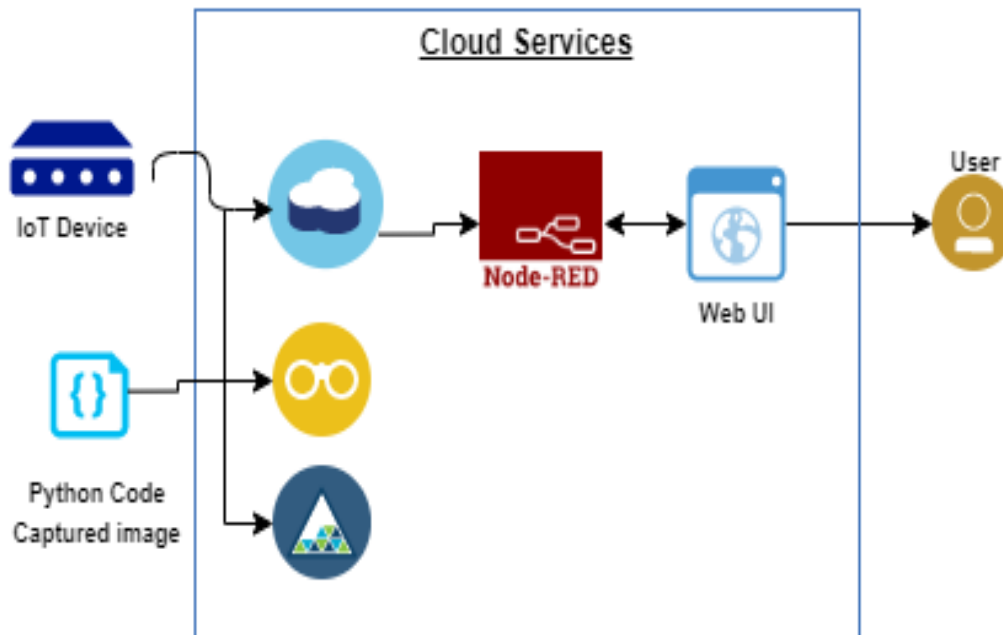
#### **SOFTWARE :**

- ✓ Python IDLE
- ✓ IBM IOT Platform
- ✓ Node red service
- ✓ Cloud Database
- ✓ Cloud Object Storage

## Block Diagram



### ***Flow chart:***



### ***Experimental Investigations :***

- After creating all the services ,run the program.
- Then its start detecting the person,and takes the data and upload to Cloudant DB.
- It also stores the data(images) in the cloud object storage.
- Where the user can access those url and the images.
- We have created the node flow and the user inter face.
- With the help of text to speech voice commands are generated.
- After the output the user can visual the data.

## RESULT :

The image displays a screenshot of a Windows desktop with two applications open: Node-RED and IDLE Shell 3.9.5.

**Node-RED Interface:**

- Flow 3:** The main workspace shows a workflow. It starts with an **IBM IoT** node (connected), followed by a **function** node. The output of the function node goes to a **msg.payload** node. This is then connected to a **switch** node, which leads to an **rbe** node, and finally to an **http request** node. Another **http request** node is connected to a **function** node, which then connects to a **template** node. A **click here** node is also connected to the **http request** node.
- Left Panel:** Shows the **common** and **function** node palettes.
- Right Panel:** Shows the **dashboard** tab with a **Layout** section and a **tabs & links** section. A message at the bottom states: "There is 1 widget not in a group. Click here to create the missing groups".

**IDLE Shell 3.9.5:**

- File:** `Final.py` (C:\Users\Kuduchalla shekhar\OneDrive\Desktop\gcet-opencv\Final.py (3.9.5))
- Code:** The code is a Python script. It starts with a `cv2.imshow` call, followed by a `draw` function. The `draw` function is a loop that updates the `safety` and `expression` values. It then calls `cv2.imshow` and `cv2.waitKey`. The script ends with `cv2.destroyAllWindows()`.
- Output:** The output shows a `PermissionError: [Errno 13] Permission denied: 'alert.mp3'` message.

access

All Documents

Query

Permissions

Changes

Design Documents

Document ID

Options

{ } JSON

Table

Metadata

{ } JSON

Create Document

	id	key	value
<input type="checkbox"/>	3265443a62412beb53c5f62651021...	3265443a62412beb53c5f62651021...	{ "rev": "2-6e6bb3cfef3aa76008cacff...
<input type="checkbox"/>	3e4af50cde4126db7153e131d48c8...	3e4af50cde4126db7153e131d48c8...	{ "rev": "1-29a82cd9cf12e46550566...
<input type="checkbox"/>	4d75174ebe61c6077c7b584a2fc49...	4d75174ebe61c6077c7b584a2fc49...	{ "rev": "1-b2c616e3f9e602a6f2afee...
<input type="checkbox"/>	4d75174ebe61c6077c7b584a2fc49...	4d75174ebe61c6077c7b584a2fc49...	{ "rev": "1-b2c616e3f9e602a6f2afee...
<input type="checkbox"/>	4d75174ebe61c6077c7b584a2ff5a...	4d75174ebe61c6077c7b584a2ff5a...	{ "rev": "1-9098f84f46a29dfb7c1c6e...
<input type="checkbox"/>	67ceb3c5950329cf111a6bcc8bab0...	67ceb3c5950329cf111a6bcc8bab0...	{ "rev": "1-29a82cd9cf12e46550566...
<input type="checkbox"/>	68ad91401dd34e3c2a193413f0b79...	68ad91401dd34e3c2a193413f0b79...	{ "rev": "1-b2c616e3f9e602a6f2afee...
<input type="checkbox"/>	70959f21dfd03b26c86ad5bfd1b878...	70959f21dfd03b26c86ad5bfd1b878...	{ "rev": "1-b2c616e3f9e602a6f2afee...
<input type="checkbox"/>	70959f21dfd03b26c86ad5bfd1e83a...	70959f21dfd03b26c86ad5bfd1e83a...	{ "rev": "1-9098f84f46a29dfb7c1c6e...
<input type="checkbox"/>	72d609e057fc40c6e8ad987302ebb...	72d609e057fc40c6e8ad987302ebb...	{ "rev": "1-29a82cd9cf12e46550566...
<input type="checkbox"/>	7f2431b3e643b1daeefbc991b8f2264b	7f2431b3e643b1daeefbc991b8f2264b	{ "rev": "1-ca625a7feb741316e0b48...

Showing document 1 - 14. Documents per page: 20

IBM Cloud

Search resources and offerings...

Catalog

Docs

Support

Manage

Kuduchella Ma...

Storage / Cloud Object Storage-de / manohar

Transfers

Details

Actions...

Endpoints

Service credentials

Connections

Usage details

Plan

uploaded prior to the interruption will count towards billable storage. While the console will alert users to incomplete multipart uploads, it is encouraged to routinely check for and clear out incomplete uploads using the REST API or an SDK. [Learn more](#)

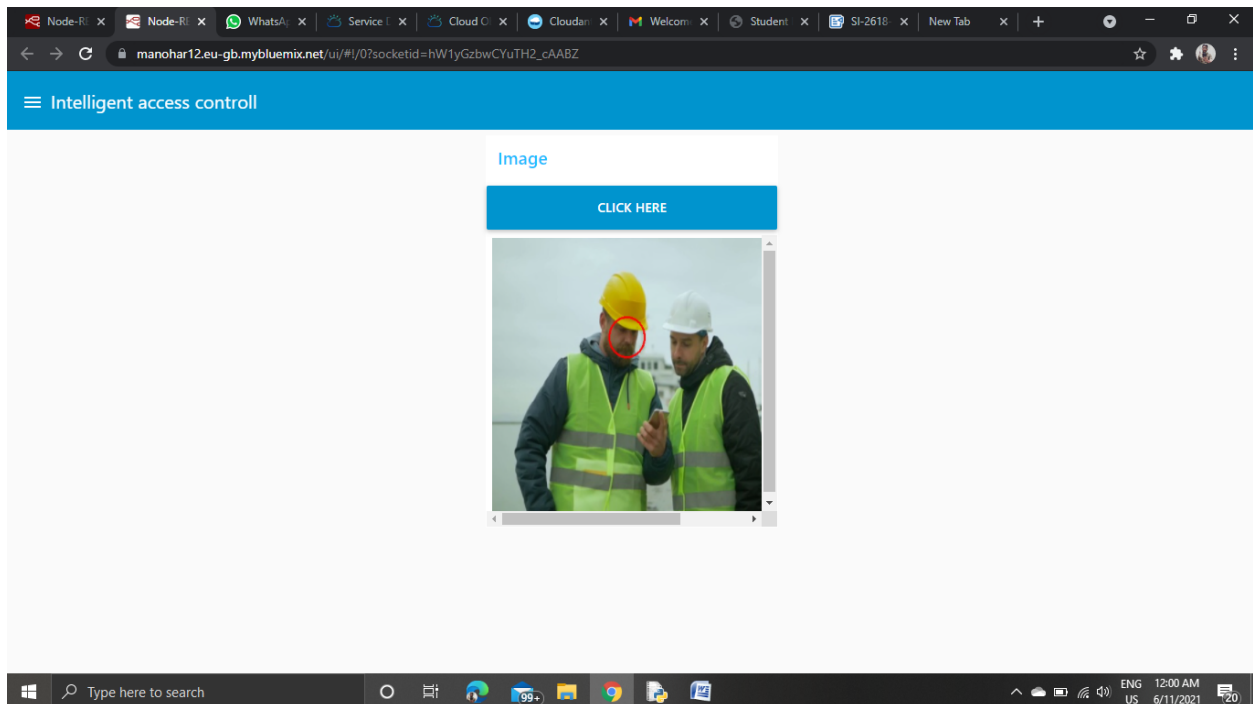
Prefix filter

Upload

<input type="checkbox"/>	Object name	Archived	Size	Last modified	
<input type="checkbox"/>	manohar/21-06-06-10-26.jpg		57.8 KB	2021-06-06 10:26 AM	
<input type="checkbox"/>	manohar/21-06-06-16-23.jpg		56.2 KB	2021-06-06 4:23 PM	
<input type="checkbox"/>	manohar/21-06-06-16-24.jpg		55.9 KB	2021-06-06 4:24 PM	
<input type="checkbox"/>	manohar/21-06-09-19-44.jpg		55.9 KB	2021-06-09 7:44 PM	

Items per page: 10 1-10 of all items page 1

Drag and drop files (objects) here or click to upload



## ***Advantages and Disadvantages :-***

### **Advantages :**

#### **INCREASE EASE OF ACCESS FOR EMPLOYEES**

An access control system allows you to “set and forget” who has access to each area of your business. Once you give the authorization, an employee can access all the areas they need to get their jobs done. With the scan of a key card or input of a PIN, the employee can get to wherever they need with ease.

#### **GET RID OF TRADITIONAL KEYS**

The use of traditional keys has a few drawbacks. Restricting access to particular areas requires individual keys. The larger the building, the more locks you need. For an individual like a janitor or a high-clearance individual, this can mean a bulky key ring and confusion about which keys do what. An access control system saves time for those accessing restricted areas and also saves you visits from the locksmith.

Also, keys can be duplicated, leaving you vulnerable to unauthorized access. If an employee

doesn't turn in their key before they leave your company, you leave yourself unprotected or must get your locks changed. Access control security does away with this.

### **SAVE MONEY AND ENERGY**

With access control security, you save money on locks and security personnel. An access control system can verify a person's identity without the need for a security guard.

Access control systems can also be integrated with lighting, heating and cooling systems. Lights can turn on when there are people in a room and will shut off when they leave. You can also adjust temperatures when no one is in an area to save on energy costs.

### **KEEP TRACK OF WHO COMES AND GOES**

An access control system gives you data on who enters and exits a building or room and when. You can ensure people are working when they are supposed to be. If theft or an accident occurs, you know exactly who accessed a specific area at the time of the incident.

### **PROTECT AGAINST UNWANTED VISITORS**

A large company creates an opportunity for visitors to go undetected. One of the benefits of using access control systems is that unauthorized people cannot get in. Since doors need credentials before they unlock, only those you've given credentials to can access the area. With this system, you can be sure everyone in your building is supposed to be there, whether you know them or not.

### **COMPLY WITH INDUSTRY REGULATIONS OR SECURITY STANDARDS**

There are many regulations for data security that need restrictions on physical access to data. Anyone in the healthcare industry must comply with HIPPA, but most organizations are subject to these regulations as well. If an employee or student requests a medical leave, records of the individual's illness must be kept secure. In the world of commerce, customers' financial information must be kept safe, and IT Departments must restrict access to servers and digital data.

All this information, whether it is a digital or paper file, can be secured with an access control system.

## **REDUCE THEFT AND ACCIDENTS**

You can protect your company's assets, expensive equipment or even office supplies by controlling access. You can restrict access to supply closets and computer banks, so only trusted individuals can access them. Employees know their arrivals and departures are tracked, which deters theft.

Also, lab equipment or chemicals in schools or hospitals can injure people who aren't trained to use them. To prevent accidents, you can restrict access to only those who know how to follow safety protocols.

## **PREVENT AGAINST DATA BREACHES**

Health information, financial records and client data are often stored on company-owned servers. Access control systems can restrict or grant access to IT rooms and even individual computers or networks, so only trusted individuals may access them.

### **Disadvantage:**

#### **Hacking**

Access control systems can be hacked. When a system is hacked, a person has access to several people's information, depending on where the information is stored. [Wired](#) reported how one hacker created a chip that allowed access into secure buildings, for example. Not only does hacking an access control system make it possible for the hacker to take information from one source, but the hacker can also use that information to get through other control systems legitimately without being caught. Despite access control systems increasing in security, there are still instances where they can be tampered with and broken into.



### ***Applications :***

Can be used in special areas like Airports, Port Trusts, Naval Base, Military base, Telecom buildings, Refinery, Petroleum handling areas etc. needed highly sophisticated security and access control system.

### ***Conclusion :***

We have came up with an access system ,which provide a high level of security for the user.

### ***Future scope :***

Access from anywhere makes a remote or flexible workforce easier to manage as well. To accommodate changes in company growth, remote onboarding and offboarding offers a better employee experience, as well as maintains workplace security. In addition, being able to containerize space, add and remove entries, and adjust permissions on the fly will keep organizations from wasting resources.

### ***Bibliography :***

- Smart Internz mentors
- Smart Internz lecture videos
- Smart Internz material

## Appendix :

```
Final.py - C:\Users\Kuduchalla shekhar\OneDrive\Desktop\gcet-opencv\Final.py (3.9.5)
File Edit Format Run Options Window Help

import cv2
import time
import datetime
import ibm_boto3
from ibm_botocore.client import Config, ClientError
from ibmcloudant.cloudant_v1 import CloudantV1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator
import wiotp.sdk.device

# Constants for IBM COS values
COS_ENDPOINT = "https://manohar.s3.jp-tok.cloud-object-storage.appdomain.cloud" # Current list available at https://control.cloud-object-storage.cloud.ibm.com/v2/endpoi
COS_API_KEY_ID = "7f7b5a0849212qoG0BII-SufaeQ0-3-1c-1xIRZ0yget" # eg "W00YiXXXXXXXXXXMB-odB-2ySTrFBIQOWano--F3byk"
COS_INSTANCE_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/94a9058de7634a75b86a721e2524a041:9eaeda1b-4ff4-4fa1-94a8-9917e79c2fc3::"

# Create resource
cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)

authenticator = BasicAuthenticator('apikey-v2-2y8twpk3cni02ngsc297oqatoulogdt961768upuw79q', '43c3fef46c4ca6560b7359d05c4c3d57')
service = CloudantV1(authenticator=authenticator)
service.set_service_url('https://apikey-v2-2y8twpk3cni02ngsc297oqatoulogdt961768upuw79q:43c3fef46c4ca6560b7359d05c4c3d57@7b88ba8a-383b-49c5-ba00-9a03115de98a-bluemix.c

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)

myConfig = {
    "identity": {
        "orgId": "sn7dml",
        "typeId": "ESP32",
        "deviceId": "12345699"
    },
    "auth": {
        "token": "9390569334"
    }
}

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)

}

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

bucket = "manohar"
def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for (0) to bucket: (1)\n".format(item_name, bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5

        # set threshold to 15 MB
        file_threshold = 1024 * 1024 * 15

        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )

        print("Transfer for (0) Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: (0)\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: (0)".format(e))

import numpy as np
import cv2
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import Service_pb2_grpc
stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())
from clarifai_grpc.grpc.api import service_pb2, resources_pb2

Ln 152 Col 0
Type here to search
ENG 11:53 PM 6/10/2021
```

Final.py - C:\Users\Kuduchalla shekhar\OneDrive\Desktop\gget-opencv\Final.py (3.9.5)

File Edit Format Run Options Window Help

```
import numpy as np
import cv2
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())
from clarifai_grpc.grpc.api import service_pb2, resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2
# This is how you authenticate.
metadata = (('authorization', 'Key 3d106b4dd8784d9786a4adde81b97fa9'),)

cap = cv2.VideoCapture('worker.mp4')
if(cap.isOpened()==True):
    print('File opened')
else:
    print('File not found')

while(cap.isOpened()):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    imS = cv2.resize(frame, (960, 540))
    cv2.imwrite('ex.jpg', imS)
    with open("ex.jpg", "rb") as f:
        file_bytes = f.read()
    # This is the model ID of a publicly available General model. You may use any other public or custom model ID.
    request = service_pb2.PostModelOutputsRequest(
        model_id='aaa03c23b3724a16a56b629203edc62c',
        inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Image(base64=file_bytes))
    )])
    response = stub.PostModelOutputs(request, metadata=metadata)
    if response.status_code != status_code_pb2.SUCCESS:
        raise Exception("Request failed, status code: " + str(response.status_code))
    print(response)
    for concept in response.outputs[0].data.concepts:
        print('%12s: %.2f' % (concept.name, concept.value))
        if(concept.value>0.86):
            print(concept.name)
            if(concept.name == "safety"):
                from ibm_watson import TextToSpeechV1
                from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
                from playsound import playsound
                authenticator = IAMAuthenticator('y4D8ClOFi_1fJU3ezHwn_I61gsEg0C_lvL4JGsV9tgo')
                text_to_speech = TextToSpeechV1(
                    authenticator=authenticator
```

Ln: 152 Col: 0

Type here to search

Final.py - C:\Users\Kuduchalla shekhar\OneDrive\Desktop\gget-opencv\Final.py (3.9.5)

File Edit Format Run Options Window Help

```
response = stub.PostModelOutputs(request, metadata=metadata)
if response.status_code != status_code_pb2.SUCCESS:
    raise Exception("Request failed, status code: " + str(response.status_code))
print(response)
for concept in response.outputs[0].data.concepts:
    print('%12s: %.2f' % (concept.name, concept.value))
    if(concept.value>0.86):
        print(concept.name)
        if(concept.name == "safety"):
            from ibm_watson import TextToSpeechV1
            from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
            from playsound import playsound
            authenticator = IAMAuthenticator('y4D8ClOFi_1fJU3ezHwn_I61gsEg0C_lvL4JGsV9tgo')
            text_to_speech = TextToSpeechV1(
                authenticator=authenticator

            text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/40fb53ff-417c-4fc4-890e-d637562a3890')
            with open('access.mp3', 'wb') as audio_file:
                audio_file.write(
                    text_to_speech.synthesize(
                        'you can enter.',
                        voice='en-US_AllisonV3Voice',
                        accept='audio/mp3'
                    ).get_result().content)
            playsound('access.mp3')
        else:
            from ibm_watson import TextToSpeechV1
            from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
            from playsound import playsound
            authenticator = IAMAuthenticator('y4D8ClOFi_1fJU3ezHwn_I61gsEg0C_lvL4JGsV9tgo')
            text_to_speech = TextToSpeechV1(
                authenticator=authenticator

            text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/40fb53ff-417c-4fc4-890e-d637562a3890')
            with open('alert.mp3', 'wb') as audio_file:
                audio_file.write(
                    text_to_speech.synthesize(
                        'access denied please keep helmet.',
                        voice='en-US_AllisonV3Voice',
```

Ln: 152 Col: 0

Type here to search

```
Final.py - C:\Users\Kuduchalla shekhar\OneDrive\Desktop\gcet-opencv\Final.py (3.9.5)
File Edit Format Run Options Window Help

text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/40fb53ff-417c-4fc4-890e-d637562a3890')
with open('alert.mp3', 'wb') as audio_file:
    audio_file.write(
        text_to_speech.synthesize(
            text='Access denied please keep helmet.',
            voice='en-US_AllisonV3Voice',
            accept='audio/mp3'
        ).get_result().content)
    playsound('alert.mp3')
cv2.imshow('frame', imS)

#drawing rectangle boundaries for the detected face
for(x,y,w,h) in faces:
    print(x,y,w,h)
    print(type(x+h/2))
    detect = True
    cv2.circle(frame, (int(x+h/2),int(y+w/2)), int(w/2), (0,0,255), 2)
    cv2.imshow('Face detection', frame)
    picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
    cv2.imwrite(picname+".jpg",frame)
    multi_part_upload(bucket, picname+'.jpg', picname+'.jpg')
    json_document={"link":COS_ENDPOINT+'/'+bucket+'/'+picname+'.jpg'}
    response = service.post_document(db="access", document=json_document).get_result()
    print(response)

myData={'Face_detect': detect}
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
print("Data published to IBM IoT platform: ",myData)
client.commandCallback = myCommandCallback
time.sleep(2)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
cap.release()
cv2.destroyAllWindows()
```

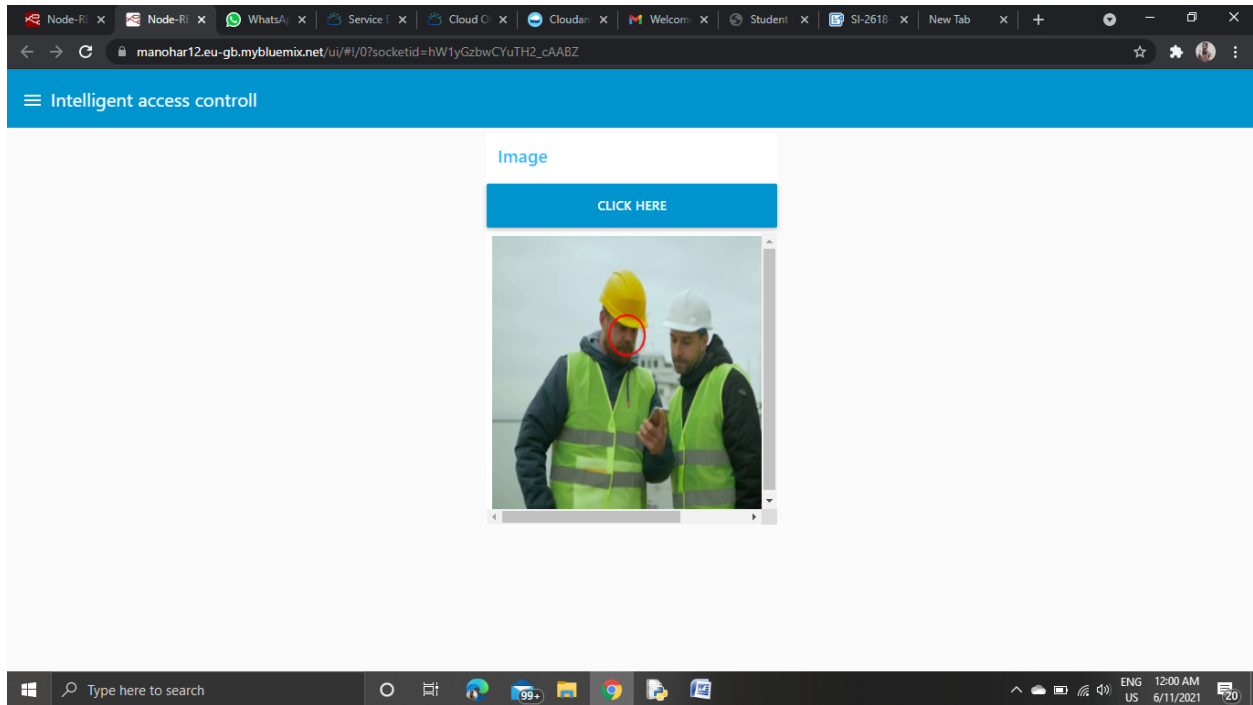


```
Final.py - C:\Users\Kuduchalla shekhar\OneDrive\Desktop\gcet-opencv\Final.py (3.9.5)
File Edit Format Run Options Window Help

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\Kuduchalla shekhar\OneDrive\Desktop\gcet-opencv\Final.py ==
2021-06-11 00:16:00.511 wiotp.sdk.device.client.DeviceClient INFO Connecte
d successfully: d:sn7dml:ESP32:1234599
File opened
Squeezed text (206 lines).
safety: 1.00
#drawl
safety
for(x,
expression: 0.99
pr
expression
pr
industry: 0.99
de
industry
cv
Traceback (most recent call last):
File "C:\Users\Kuduchalla shekhar\OneDrive\Desktop\gcet-opencv\Final.py", line
138, in <module>
with open('alert.mp3', 'wb') as audio_file:
PermissionError: [Errno 13] Permission denied: 'alert.mp3'
>>>

myData
client
print(
client
time.s
if cv2
br
cap.releas
cv2.destro
```





**THANK YOU**

**M.SATHWIK SAI -19R11A04C3**

**M.S.V SAMPATH -19R11A04C4**

**M.SAI SARATH CHANDRA-19R11A04C1**

**K.MANOHAR-19R11A04B9**

**KALVA LAHARI-19R11A04B6**